AMENDMENTS TO THE CLAIMS

Please amend claims 1, 3-5, 8, 10, 11, and 24-28 as presented below.

1. (Currently Amended) A method for deriving a process-based specification for a system, comprising:

deriving a trace-based specification from a non-empty set of traces by a processor, wherein a trace is a sequence of actions expressed as strings representing a history of an execution of a process; and

mathematically inferring the process-based specification from the trace-based specification, wherein mathematically inferring includes applying the Laws of Concurrency in reverse to a set of system traces to determine the process-based specification, wherein the process-based specification is mathematically equivalent to the trace-based specification, and whereby the Laws of Concurrency are algebraic laws that (a) allow at least one process to be manipulated and analyzed; (b) permit formal reasoning about equivalences between processes; and (c) determine traces from the at least one process.

2. (Original) The method of claim 1, wherein the process-based specification is provably equivalent to the trace-based specification.

3. (Currently Amended) The method of claim 1, further comprising:

generating the process-based specification using an inference engine, wherein the inference engine iteratively applies a set of rules to a set of data representing a problem to determine a solution to the problem by logical manipulation and analysis of the data.

4. (Currently Amended) The method of claim 3, wherein the Laws of Concurrency ~~laws of concurrency~~ are used by the inference engine to generate the process-based specification.

5. (Currently Amended) The method of claim 4, wherein the laws of concurrency are reversed by embedding the Laws of Concurrency ~~laws of concurrency~~ in the inference engine.

6. (Original) The method of claim 5, wherein the embedding is syntactic or shallow.

7. (Original) The method of claim 5, wherein the embedding is semantic or deep.

8. (Currently Amended) The method of claim 4, wherein the Laws of Concurrency ~~laws of concurrency~~ are reversed so that an equivalent process expression is output in response to a given input of at least one trace.

9. (Original) The method of claim 8, wherein multiple process expressions are given as output in response to inputs of the at least one trace.

10. (Currently Amended) The method of claim 1, further comprising:

analyzing the process-based specification to examine possible implementations of the process-based specification in different configurations, whereby analyzing includes identifying at least one equivalent alternative process-based specification and characterizing differences between the process-based specification and the at least one alternative process-based specification, wherein differences include the number of process, deterministic behavior, and competition for resources.

11. (Currently Amended) The method of claim 10, wherein the various possible implementations of the process-based specification are based on transformations of the process-based specification by applying ~~application of~~ the Laws of Concurrency ~~laws of concurrency~~ to derive various implementations.

12. (Original) The method of claim 11, wherein the various equivalent implementations are mathematically equivalent to the process-based specification.

13. (Original) The method of claim 12, wherein the various equivalent implementations are provable equivalent to the process-based specification.

14. (Original) The method of claim 13, wherein multiple correct process-based specifications are possible.

15. (Previously Amended) The method of claim 14, further comprising:

deciding which of the multiple correct process-based specifications are most appropriate.

16. (Original) The method of claim 15, wherein the process-based specification is used as a basis for generation of alternate representations.

17. (Original) The method of claim 16, wherein the alternate representations are sets of instructions.

18. (Original) The method of claim 1, wherein the set of traces is a set of sequences of events or activities specific to an application domain.

19. (Original) The method of claim 1, wherein the set of traces is derived by pre-processing a set of scenarios given as input by a user to a context sensitive editor.

20. (Original) The method of claim 19, wherein the set of scenarios is natural language text describing intended system behavior, and the elements of the set of traces are sequences of events or activities in a given application domain.

21. (Original) The method of claim 20, wherein the set of scenarios is represented by various graphical notations.

22. (Original) The method of claim 1, wherein the deriving step is repeated.

23. (Original) The method of claim 1, wherein the inferring step is repeated.

24. (Currently Amended) The method of claim 1, further comprising:

reverse engineering an existing system using the deriving step and the inferring step, whereby reverse engineering includes analyzing a process-based specification and applying the Laws of Concurrency using an inference engine having domain knowledge to infer a set of traces equivalent to the process-based specification and to derive a set of scenarios for the system equivalent to the process-based specification, wherein the scenarios are natural language text that describes the system's actions in response to incoming data and the internal goals of the system.

25. (Currently Amended) The method of claim 1, further comprising:

reverse engineering an existing system back to a set of traces using the deriving step and the inferring step, whereby reverse engineering includes analyzing a process-based specification and applying the Laws of Concurrency using an inference engine having domain knowledge to infer a set of traces equivalent to the process-based specification and to derive a set of scenarios for the system equivalent to the process-based specification, wherein the scenarios are natural language text that describes the system's actions in response to incoming data and the internal goals of the system.

26. (Currently Amended) A system adapted for deriving a process-based specification, comprising:

at least one natural language scenario;

a computer-readable medium having instructions stored thereon for deriving a trace-based specification from the at least one natural language scenario; and

an inference engine to mathematically infer the process-based specification from the trace-based specification, wherein mathematically inferring includes applying the Laws of Concurrency in reverse to a set of system traces to determine the process-based specification, wherein ~~such that~~

the process-based specification is mathematically equivalent to the trace-based specification, and whereby the Laws of Concurrency are algebraic laws that (a) allow at least one process to be manipulated and analyzed; (b) permit formal reasoning about equivalences between processes; and (c) determine traces from the at least one process.

27. (Currently Amended) A system adapted for deriving a process-based specification, comprising:

a non-empty set of traces;

a computer-readable medium having instructions stored thereon for deriving a trace-based specification from the set of traces, wherein a trace is a sequence of actions expressed as strings representing a history of an execution of a process; and

an inference engine to mathematically infer the process-based specification from the trace-based specification, wherein mathematically inferring includes applying the Laws of Concurrency in reverse to a set of system traces to determine the process-based specification, wherein ~~such that~~ the process-based specification is mathematically equivalent to the trace-based specification, and whereby the Laws of Concurrency are algebraic laws that (a) allow at least one process to be manipulated and analyzed; (b) permit formal reasoning about equivalences between processes; and (c) determine traces from the at least one process.

28. (Currently Amended) A method for deriving a process-based specification for a system, wherein the system performs actions, comprising:

receiving at least one natural language scenario describing the actions;

generating a trace-based specification from the at least one natural language scenario; and mathematically inferring the process-based specification from the trace-based specification, wherein mathematically inferring includes applying the Laws of Concurrency in reverse to a set of system traces to determine the process-based specification, wherein the process-based specification is mathematically equivalent to the actions defined above, whereby the Laws of Concurrency are algebraic laws that (a) allow at least one process to be manipulated and analyzed; (b) permit formal reasoning about equivalences between processes; and (c) determine traces from the at least one process.